09/269485

Fig.1

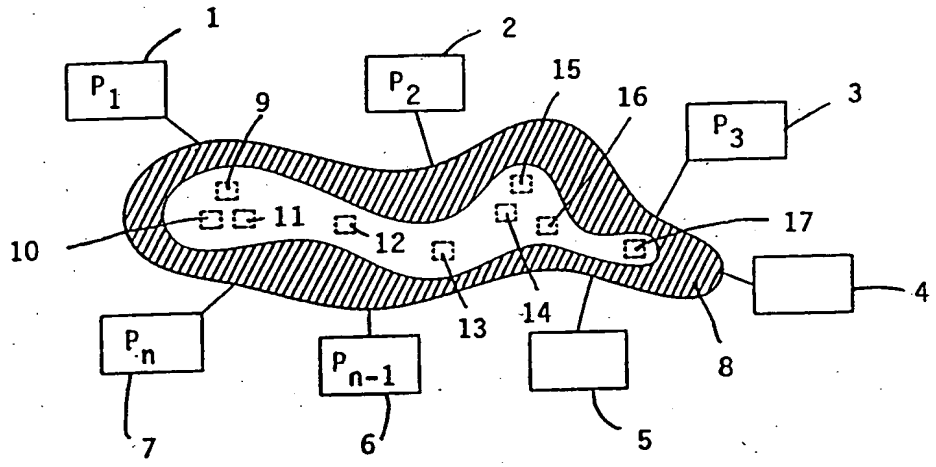


Fig.2

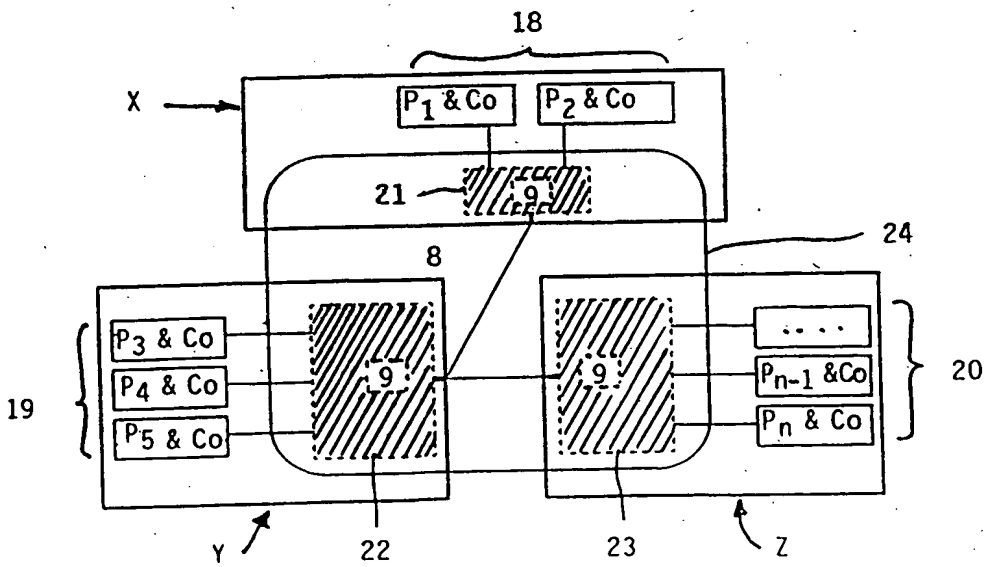
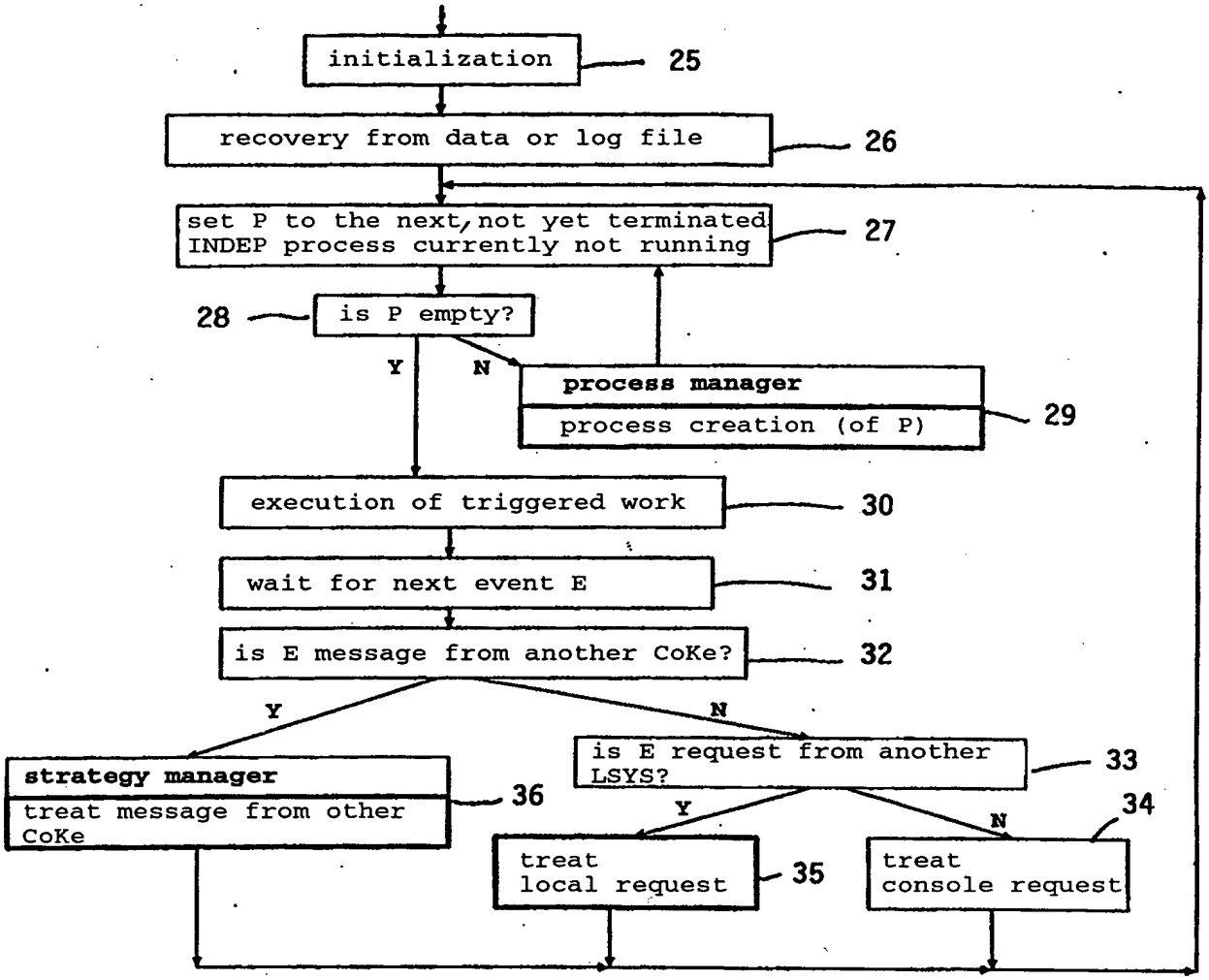


Fig.3


initialization
25
recovery from data or log file
26
set P to the next, not yet terminated INDEP process currently not running
27
28
is P empty?
Y
N
process manager
process creation (of P)
29
execution of triggered work
30
wait for next event E
31
is E message from another CoKe?
32
Y
N
strategy manager
treat message from other CoKe
36
is E request from another LSYS?
33
Y
N
34
treat local request
35
treat console request

PTO 2 9 MAR 1999

Fig.4

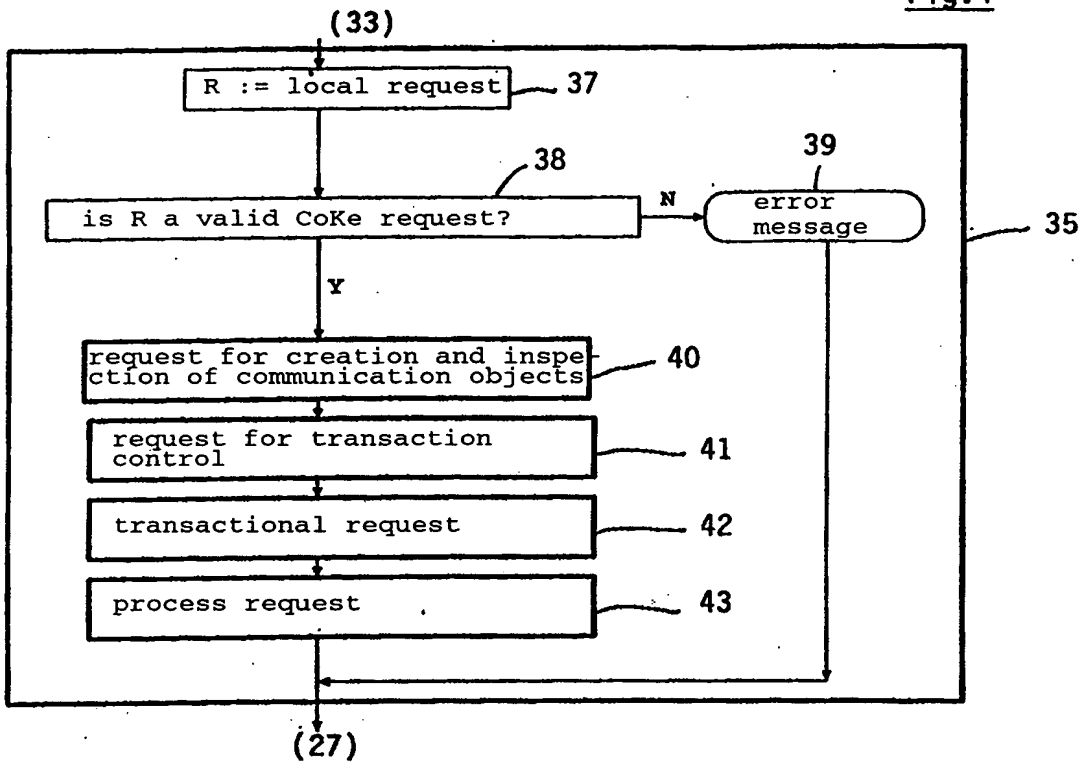
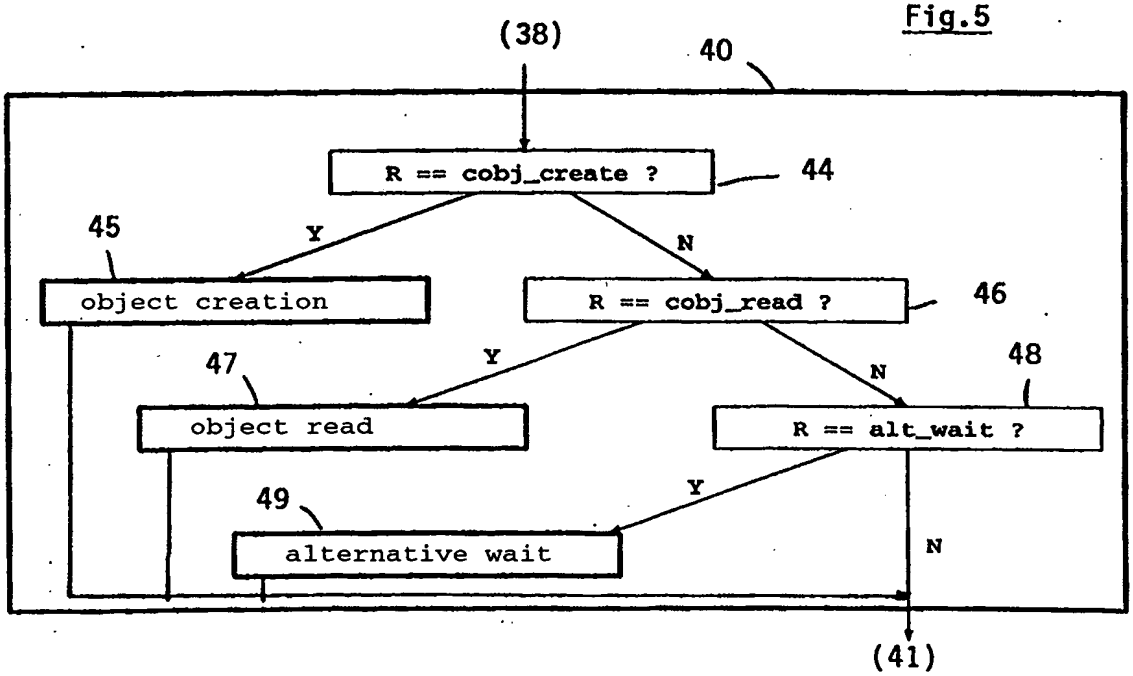


Fig.5



09/269485

RECEIVED PCT/PTO 29 MAR 1999

Fig.6

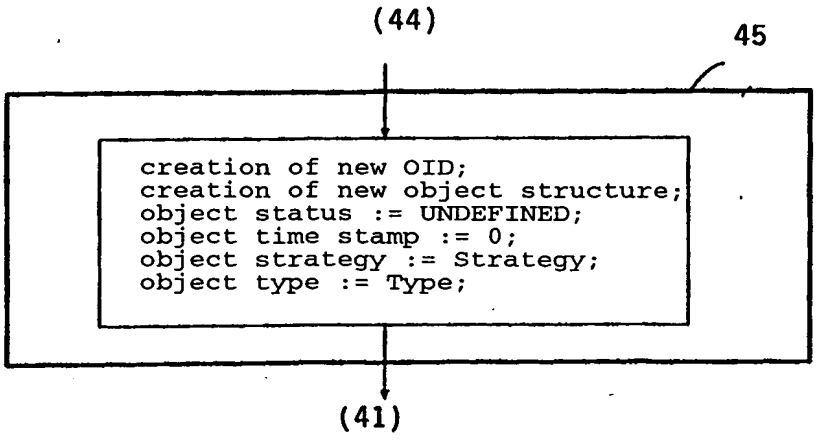
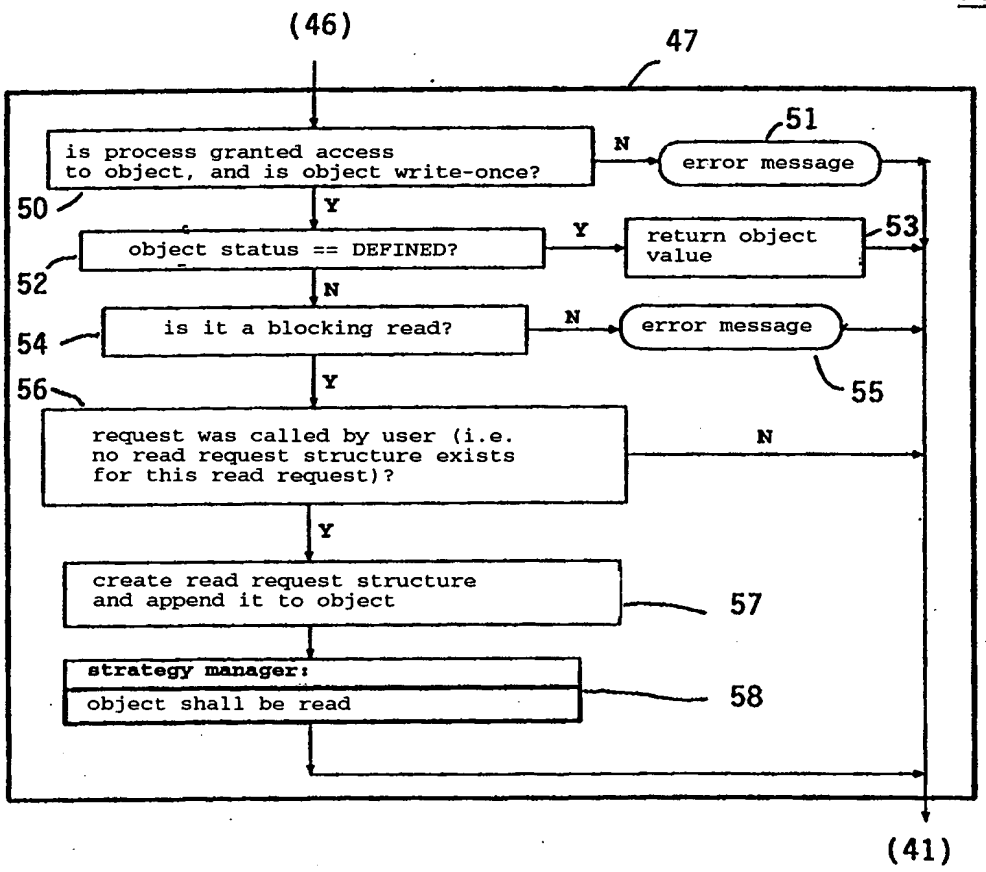


Fig.7



Rec'd PCT/PTO 29 MAR 1999

Fig.8

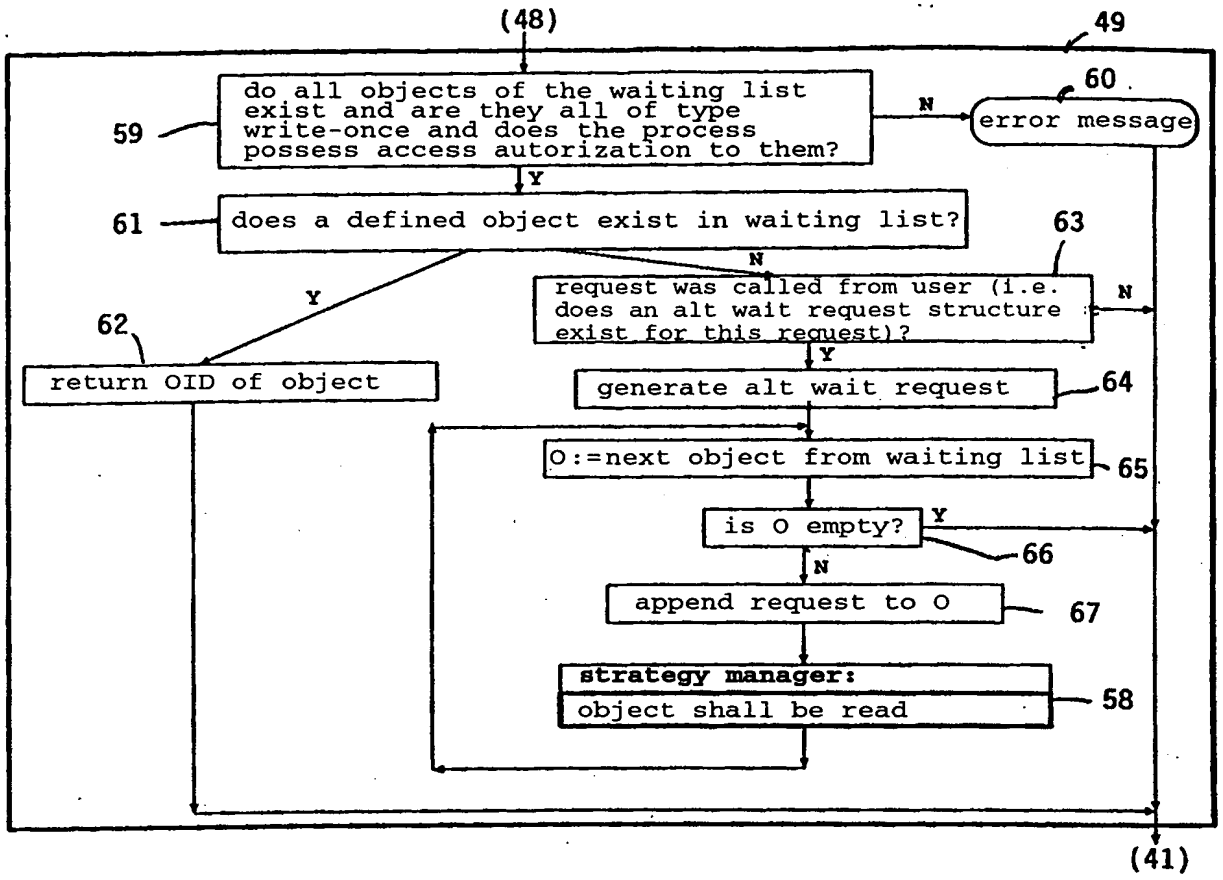


Fig.10

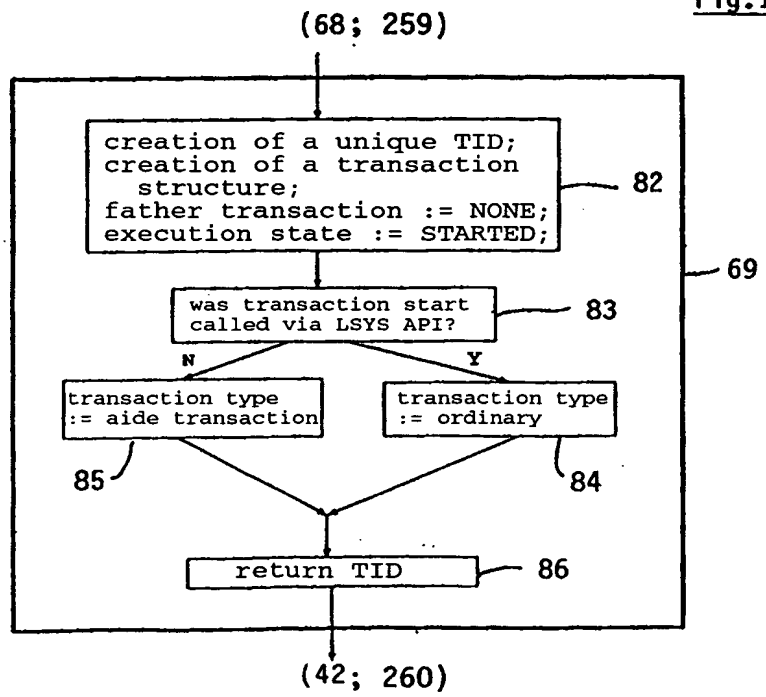


Fig.9

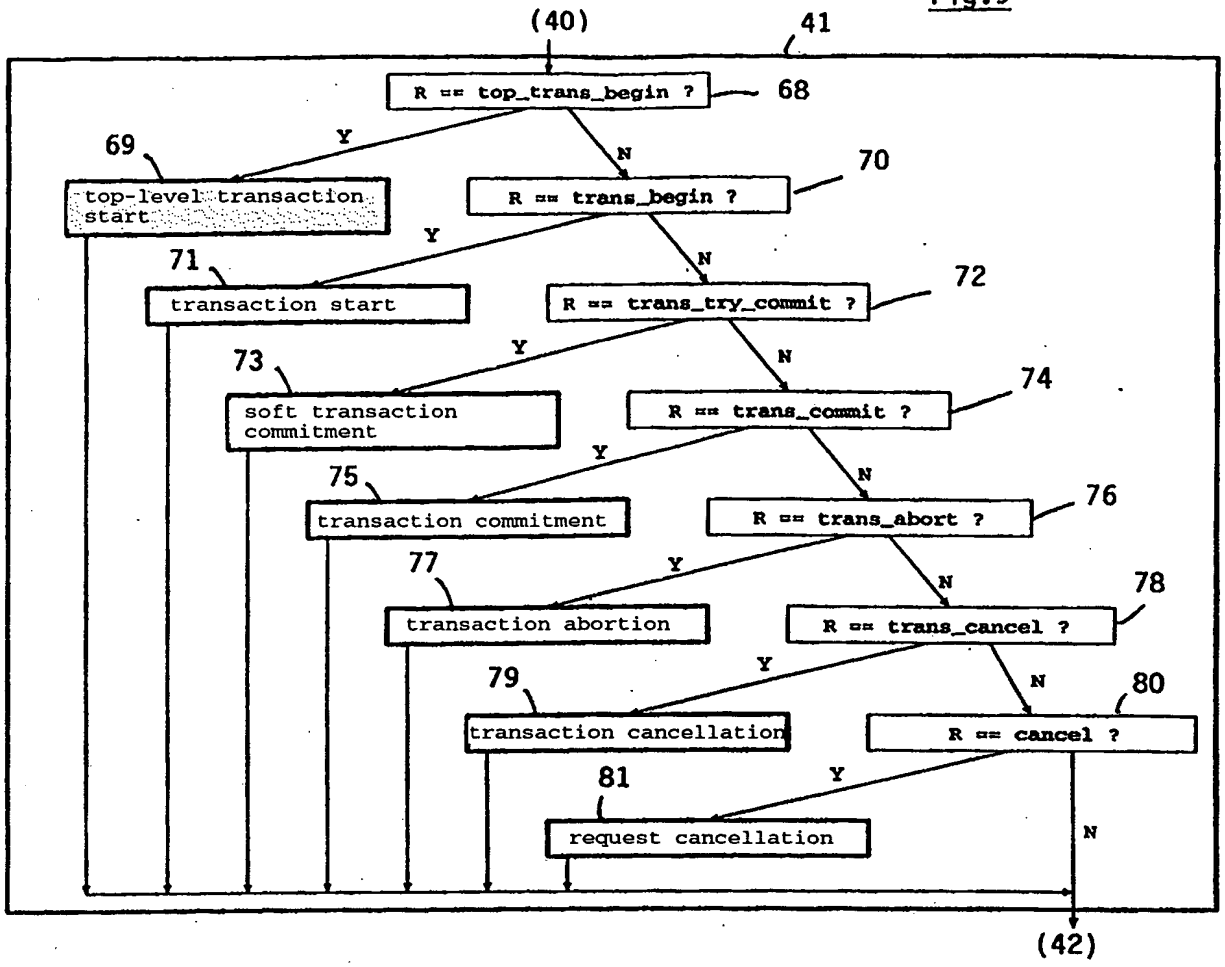
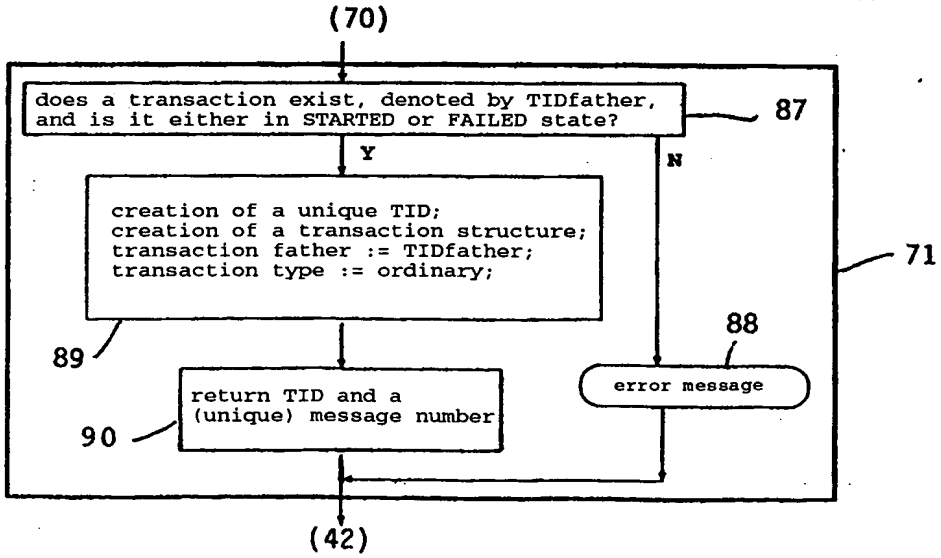
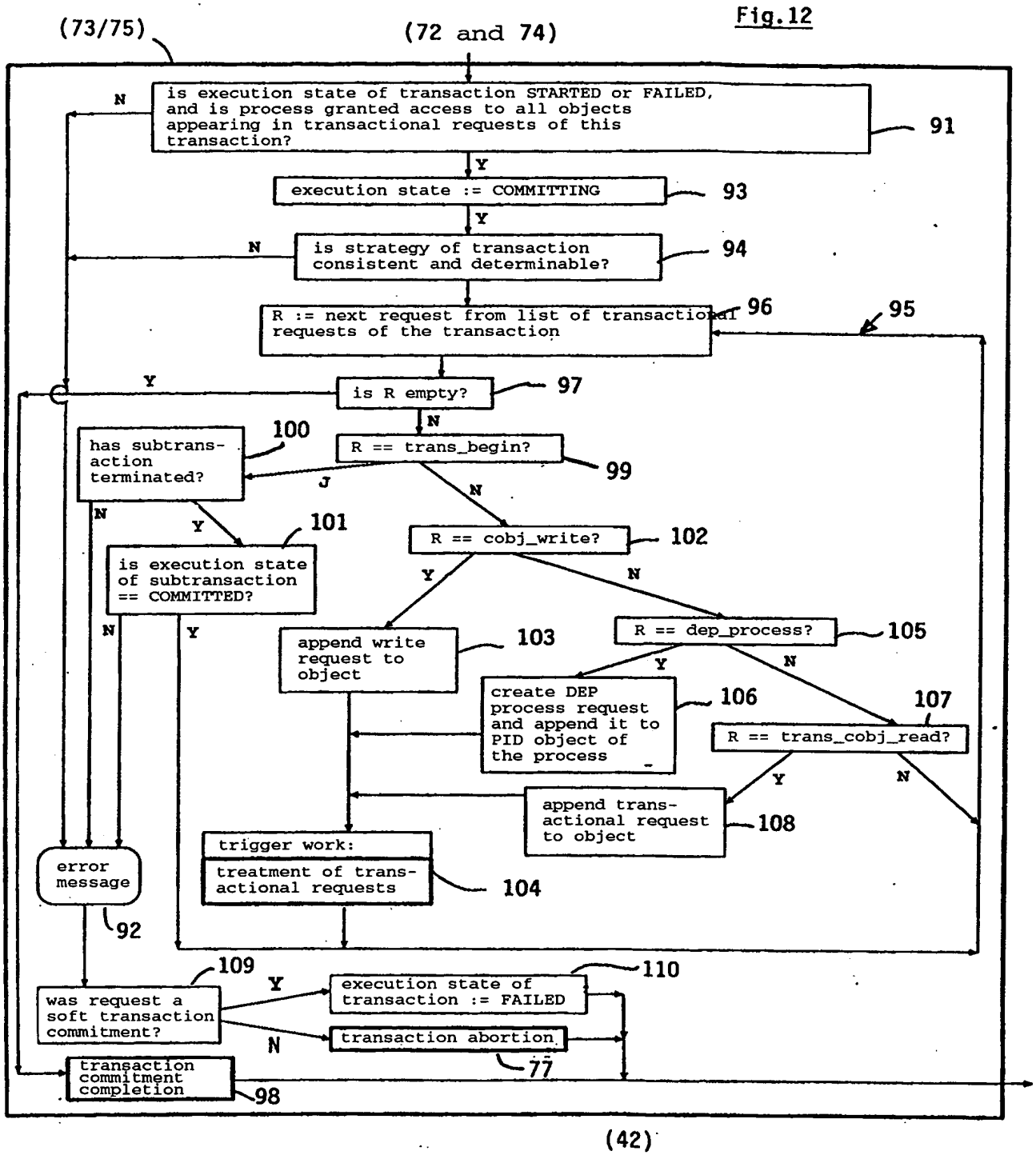

(40)
41
R == top_trans_begin ?
68
Y
N
69
top-level transaction start
R == trans_begin ?
70
Y
N
71
transaction start
R == trans_try_commit ?
72
Y
N
73
soft transaction commitment
R == trans_commit ?
74
Y
N
75
transaction commitment
R == trans_abort ?
76
Y
N
77
transaction abortion
R == trans_cancel ?
78
Y
N
79
transaction cancellation
R == cancel ?
80
81
Y
request cancellation
N
(42)

Fig.11


(70)
does a transaction exist, denoted by TIDfather, and is it either in STARTED or FAILED state?
87
Y
N
creation of a unique TID;
creation of a transaction structure;
transaction father := TIDfather;
transaction type := ordinary;
71
89
88
error message
return TID and a (unique) message number
90
(42)

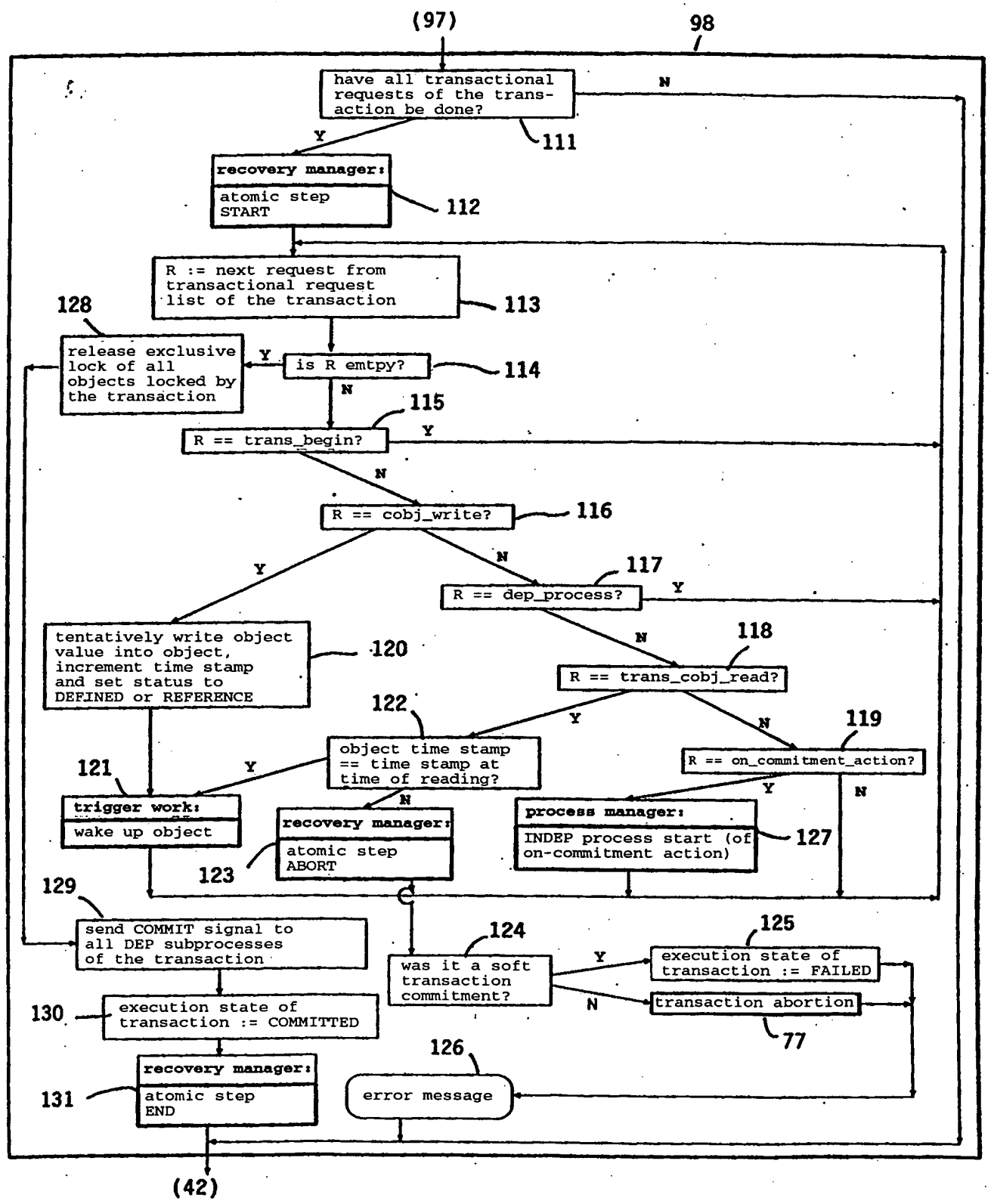
PCT/EPTO 29 MAR 1999

Fig.12



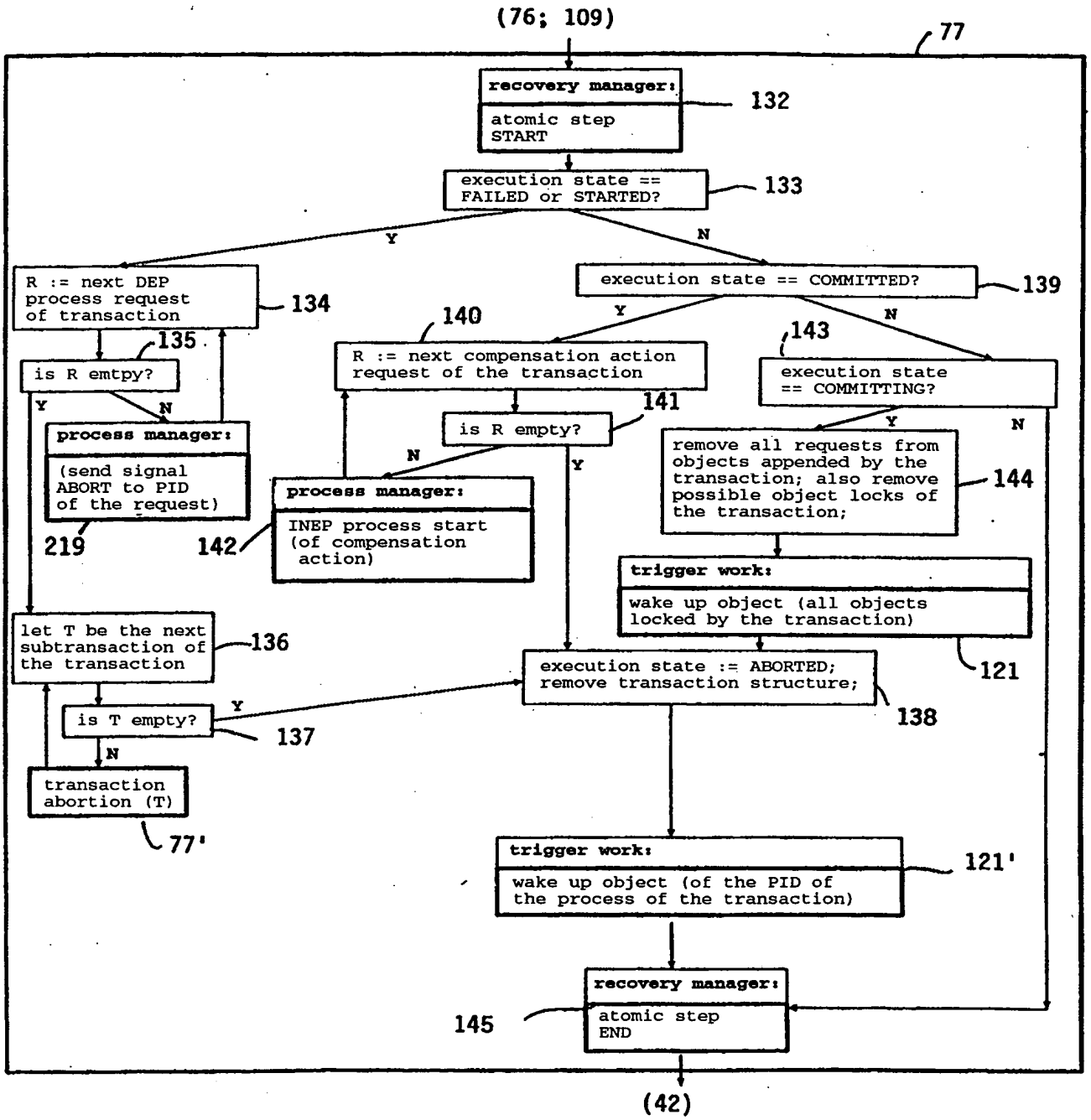
09/269485

Fig.13



09/269485

Fig.14



PCT/PTO 2 9 MAR 1999

Fig.15

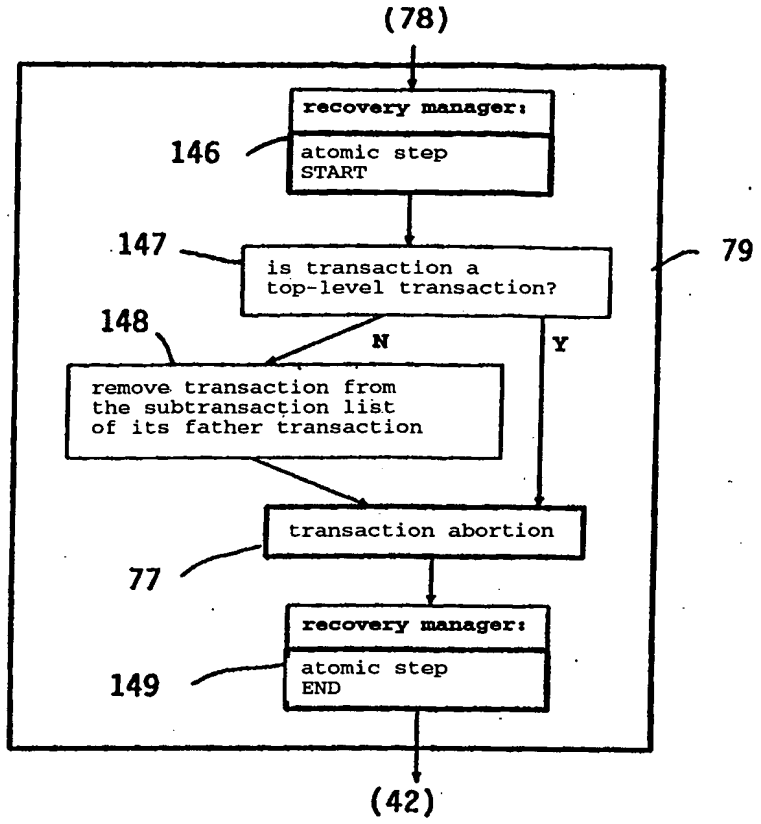


Fig.16

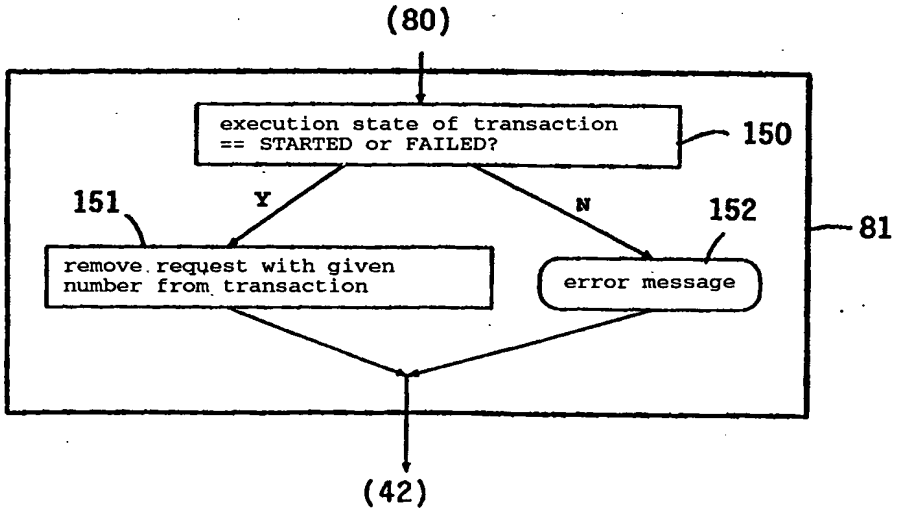


Fig.17

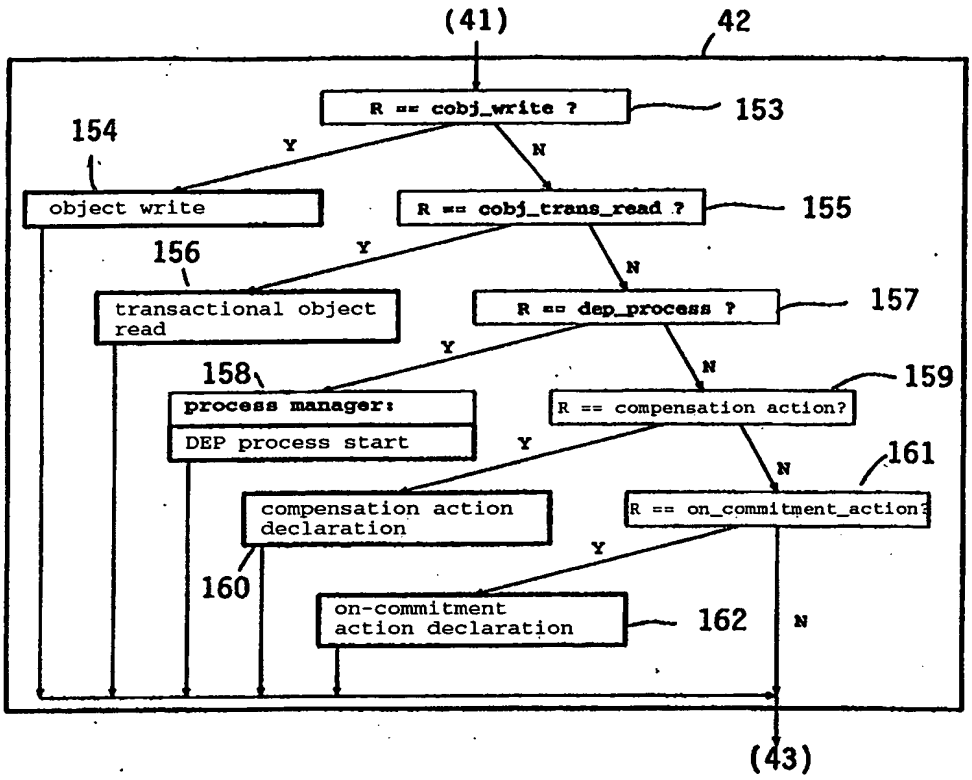
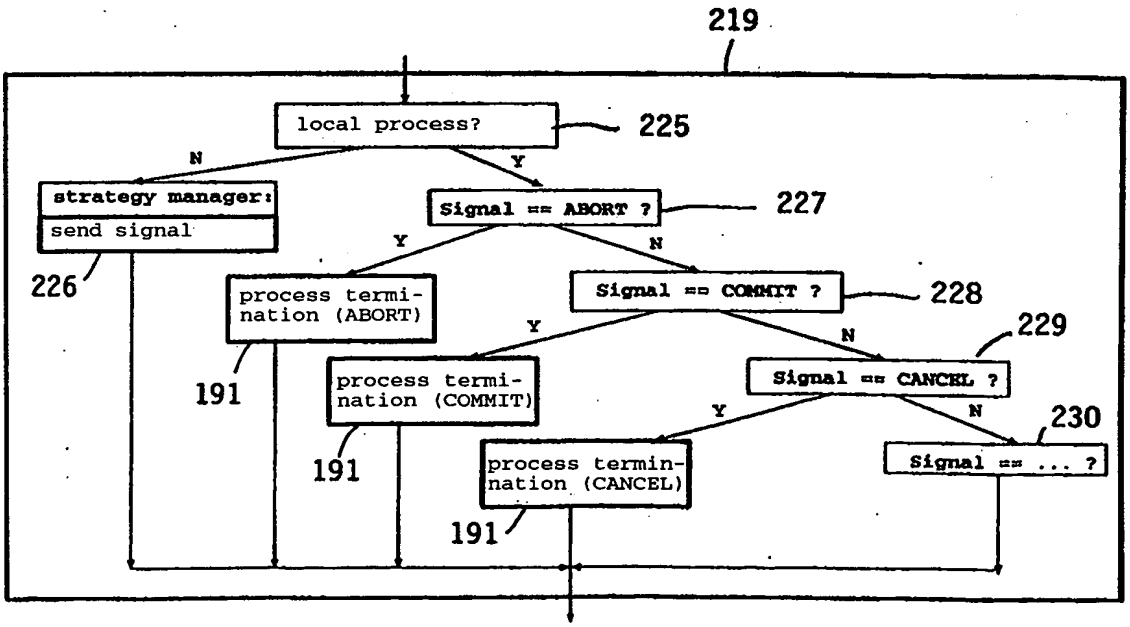

(41)
42
R == cobj_write ?
153
154
Y
N
object write
R == cobj_trans_read ?
155
156
Y
N
transactional object read
R == dep_process ?
157
Y
N
158
process manager:
DEP process start
R == compensation action?
159
Y
N
161
compensation action declaration
R == on_commitment_action?
160
Y
on-commitment action declaration
162
N
(43)

Fig.27


219
local process?
225
N
Y
strategy manager:
send signal
Signal == ABORT ?
227
226
Y
N
process termination (ABORT)
Signal == COMMIT ?
228
191
Y
N
229
process termination (COMMIT)
Signal == CANCEL ?
191
Y
N
230
process termination (CANCEL)
Signal == ... ?
191

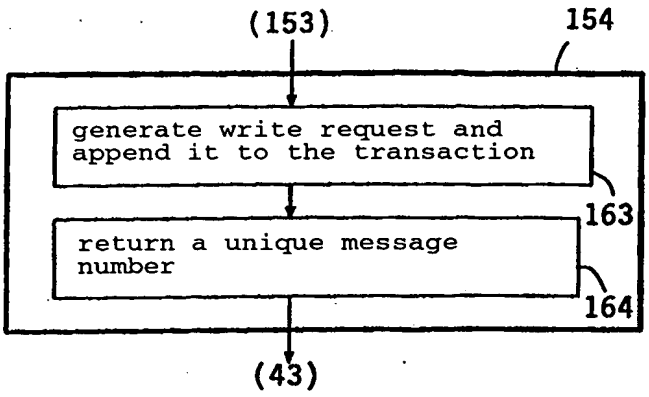
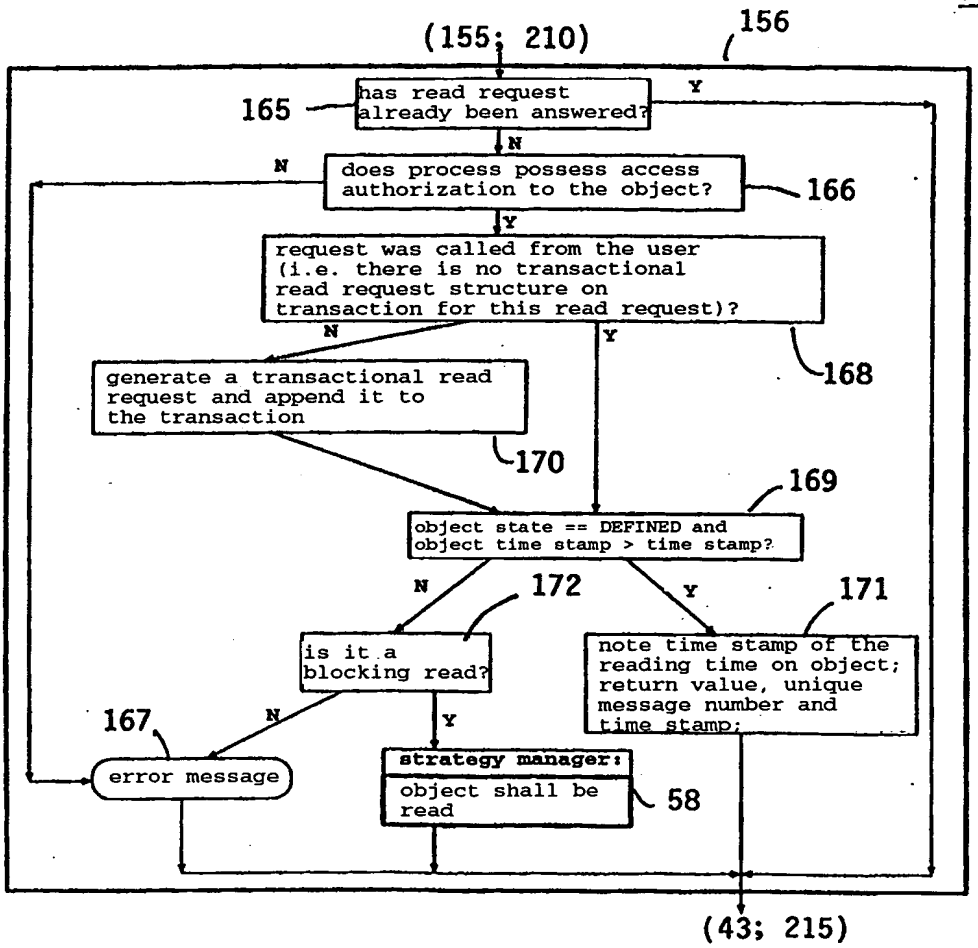

(153)
154
generate write request and append it to the transaction
163
return a unique message number
164
(43)

Fig.18

Fig.19


(155; 210)
156
165
has read request already been answered?
Y
N
N
does process possess access authorization to the object?
166
Y
request was called from the user (i.e. there is no transactional read request structure on transaction for this read request)?
N
Y
168
generate a transactional read request and append it to the transaction
170
169
object state == DEFINED and object time stamp > time stamp?
N
172
Y
171
is it a blocking read?
note time stamp of the reading time on object; return value, unique message number and time stamp;
N
167
Y
error message
strategy manager:
object shall be read
58
(43; 215)

PRI MAR 29 PTO

Fig.20

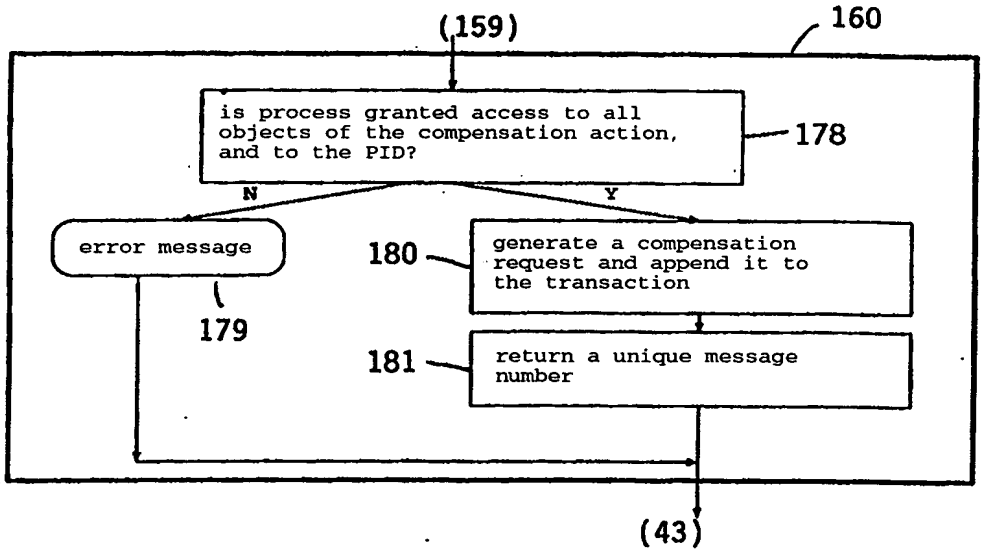


Fig.21

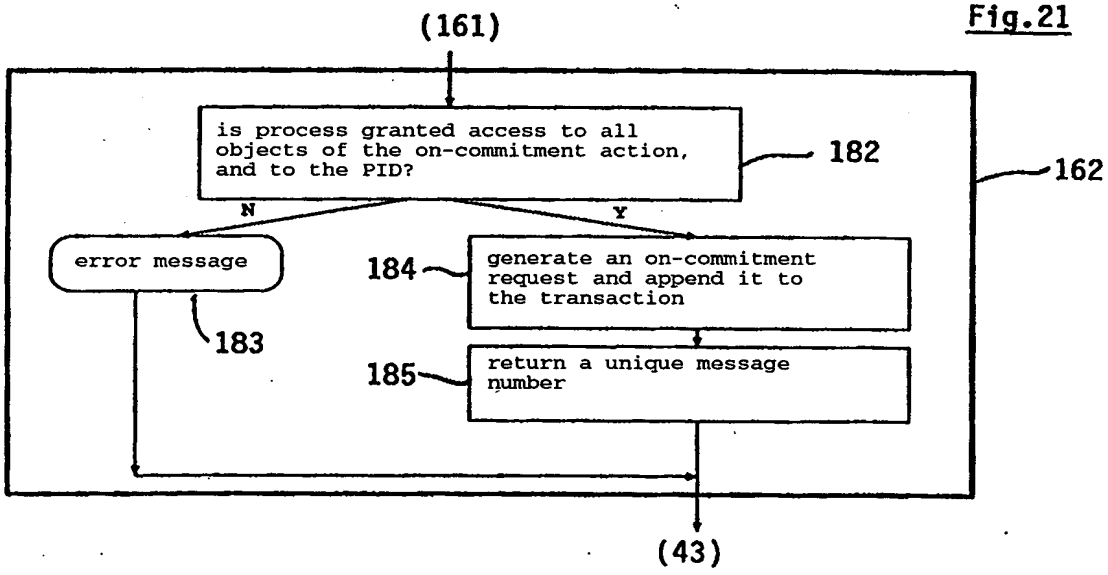


Fig.22

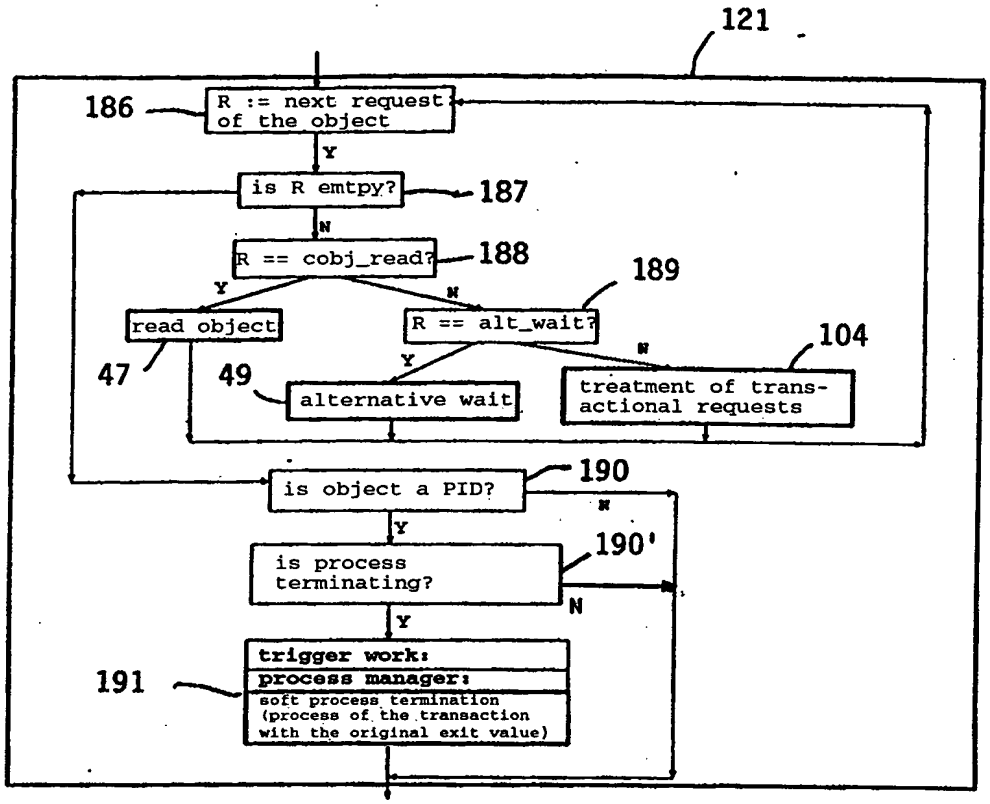
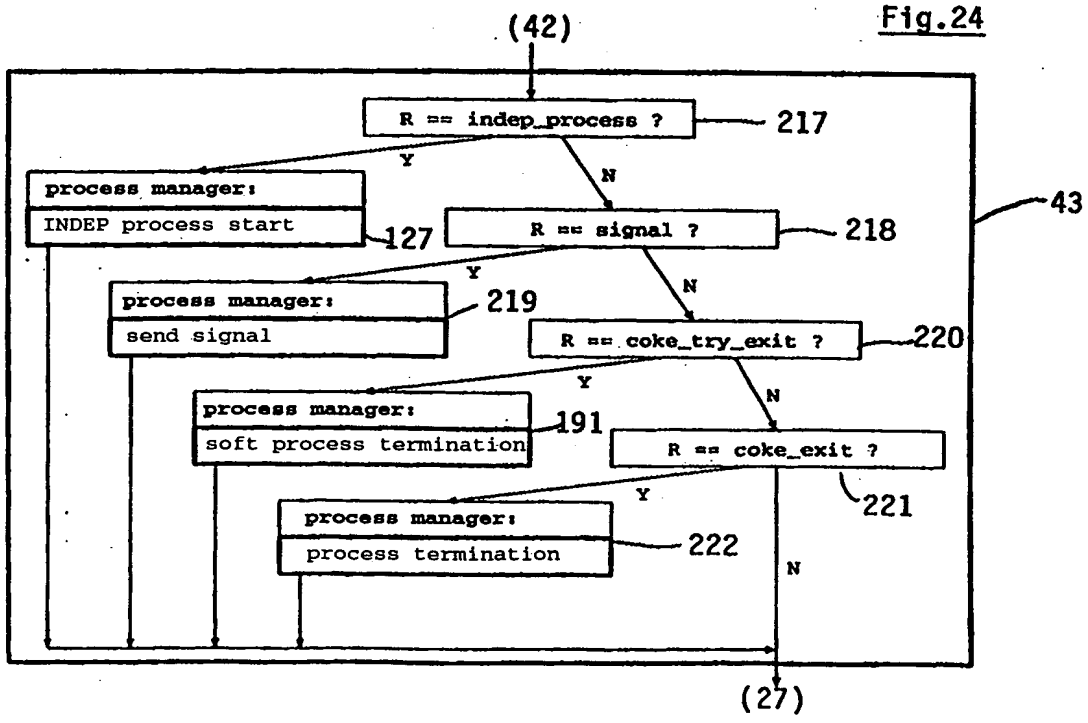
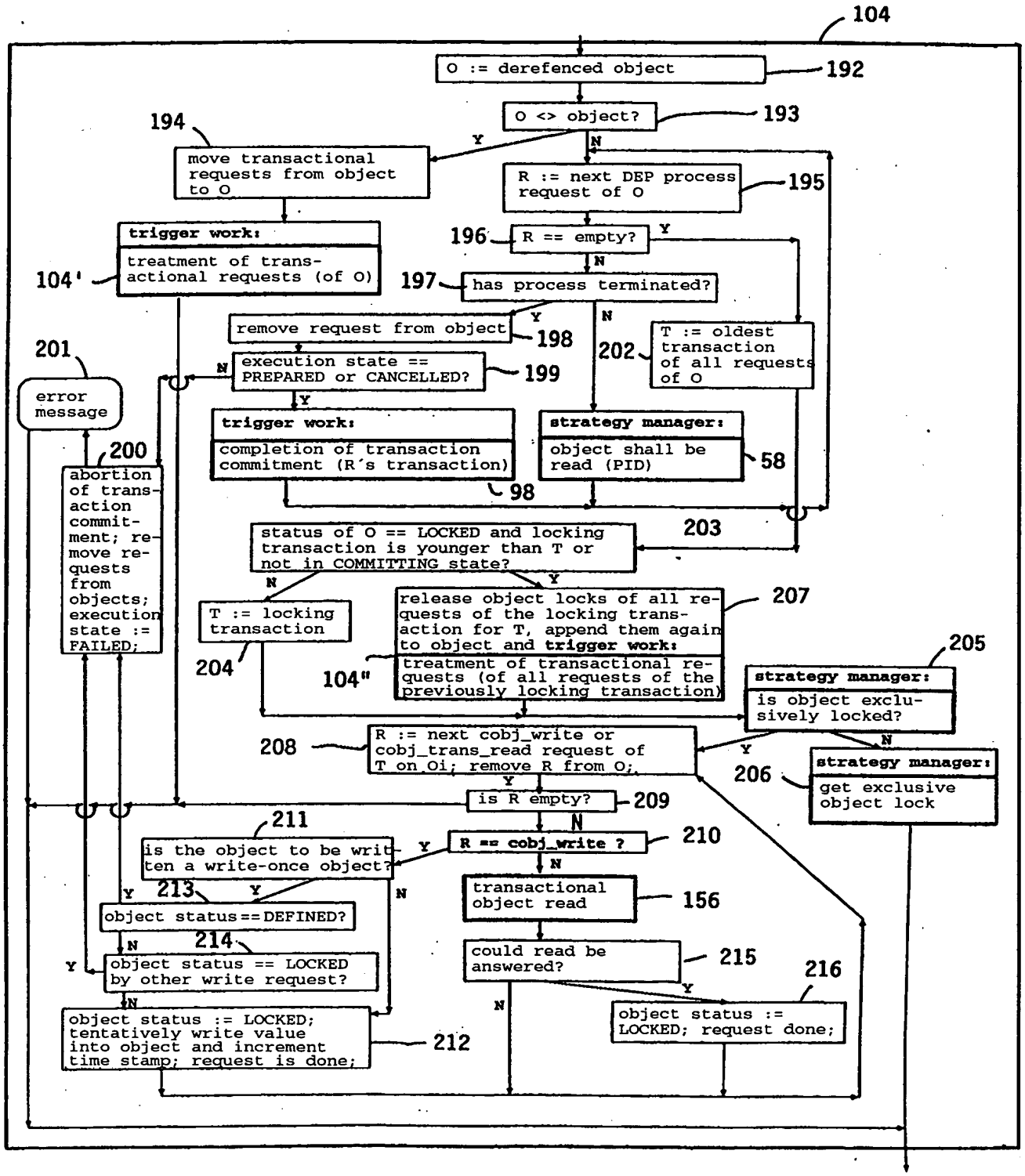

121
186
R := next request of the object
Y
is R emtpy?
187
N
R == cobj_read?
188
189
Y
N
read object
R == alt_wait?
104
47
49
Y
N
alternative wait
treatment of trans-actional requests
is object a PID?
190
N
Y
is process terminating?
190'
N
Y
trigger work:
process manager:
191
soft process termination (process of the transaction with the original exit value)

Fig.24


(42)
R == indep_process ?
217
Y
N
process manager:
INDEP process start
127
R == signal ?
218
43
Y
N
process manager:
send signal
219
R == coke_try_exit ?
220
Y
N
process manager:
soft process termination
191
R == coke_exit ?
Y
221
process manager:
process termination
222
N
(27)

09/269485

Fig.23



4/6 Rec'd PCT/PTO 29 MAR 1999

Fig.25

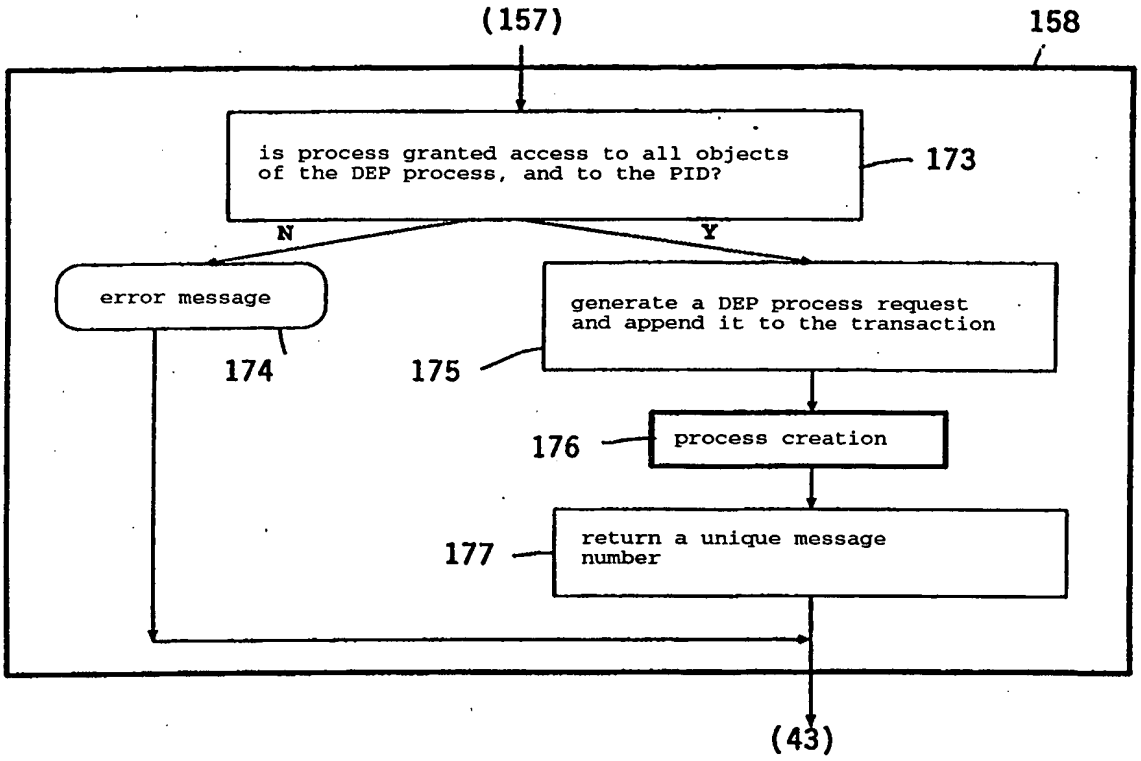
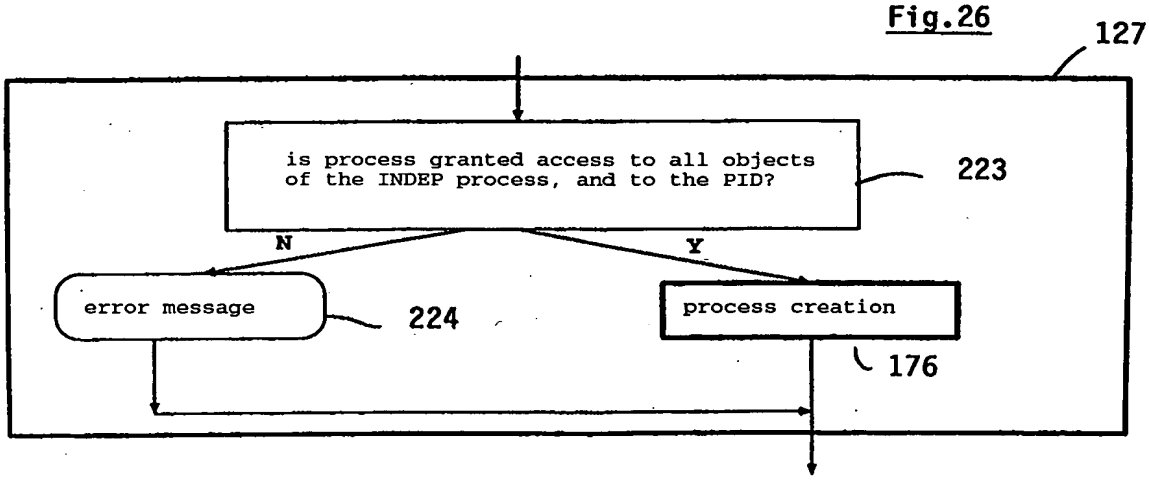


Fig.26



09/269485

Fig.28

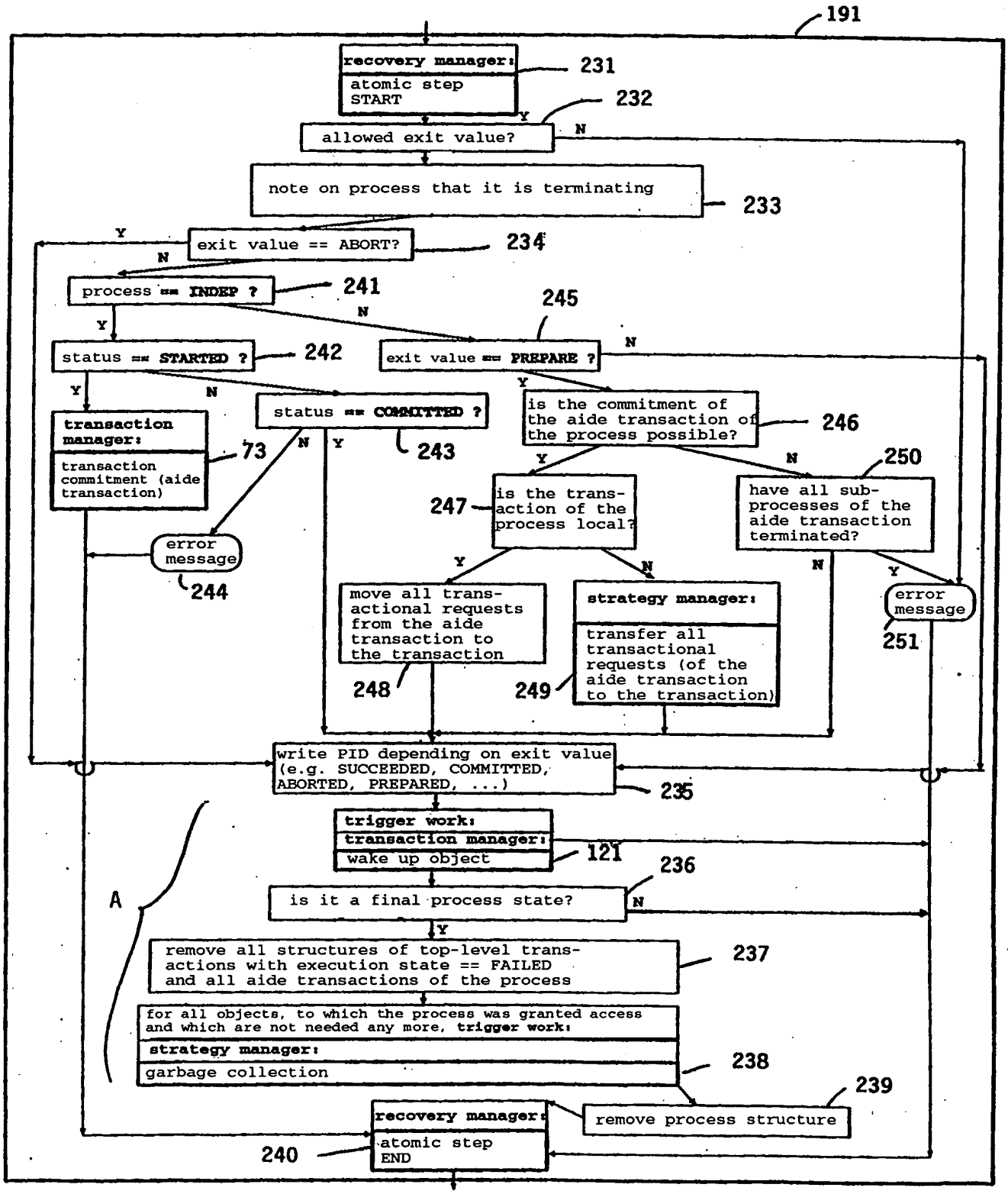


Fig.29

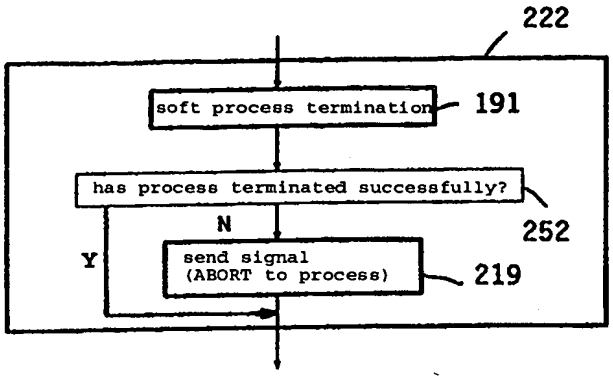
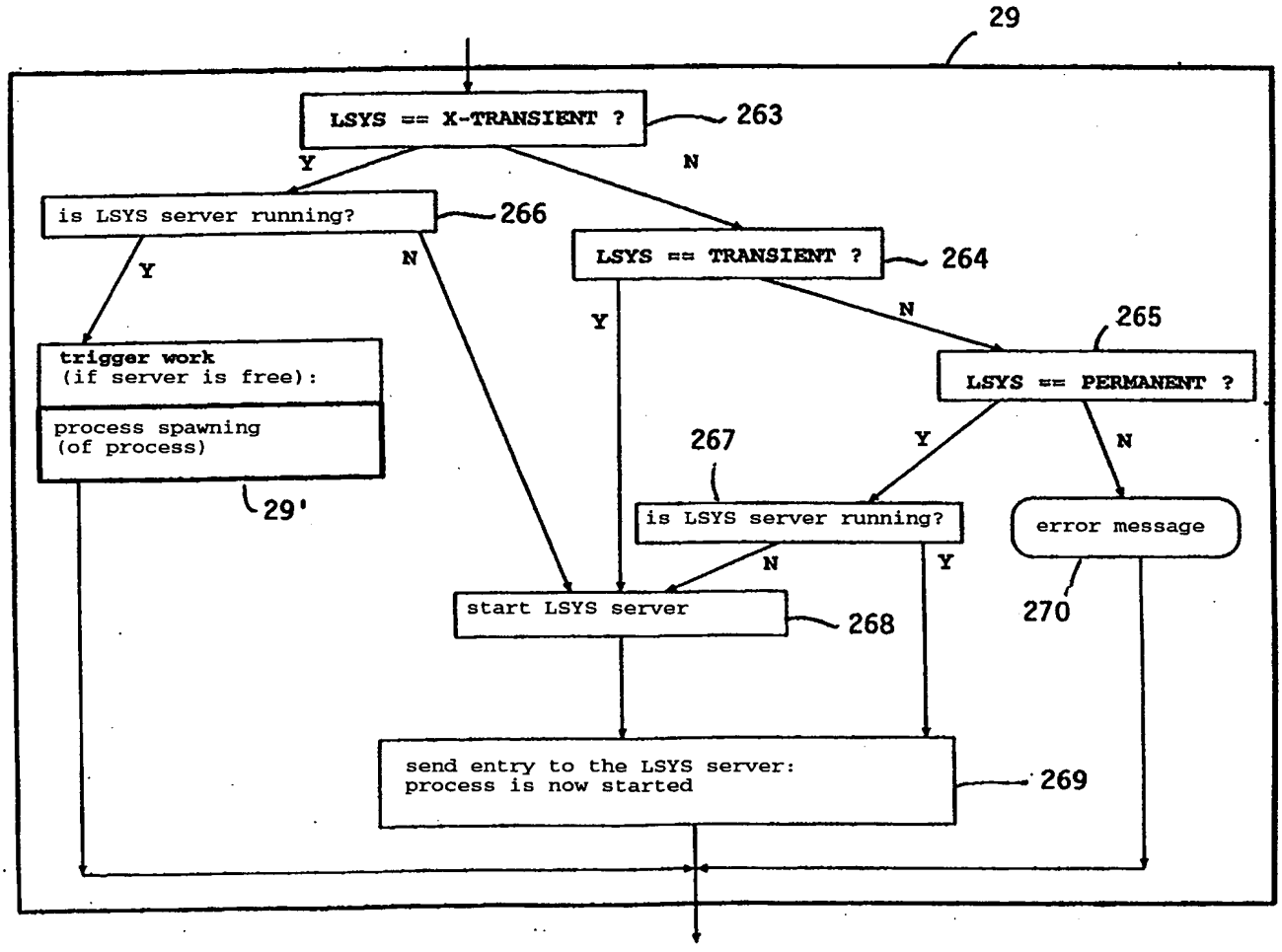

222
soft process termination
191
has process terminated successfully?
252
N
Y
send signal
(ABORT to process)
219

Fig.31


29
LSYS == X-TRANSIENT ?
263
Y
N
is LSYS server running?
266
LSYS == TRANSIENT ?
264
Y
N
Y
N
trigger work
(if server is free):
process spawning
(of process)
29'
265
LSYS == PERMANENT ?
267
Y
N
is LSYS server running?
error message
N
Y
start LSYS server
268
270
send entry to the LSYS server:
process is now started
269

416 Rec'd PCT/PTO 29 MAR 1999

Fig.30

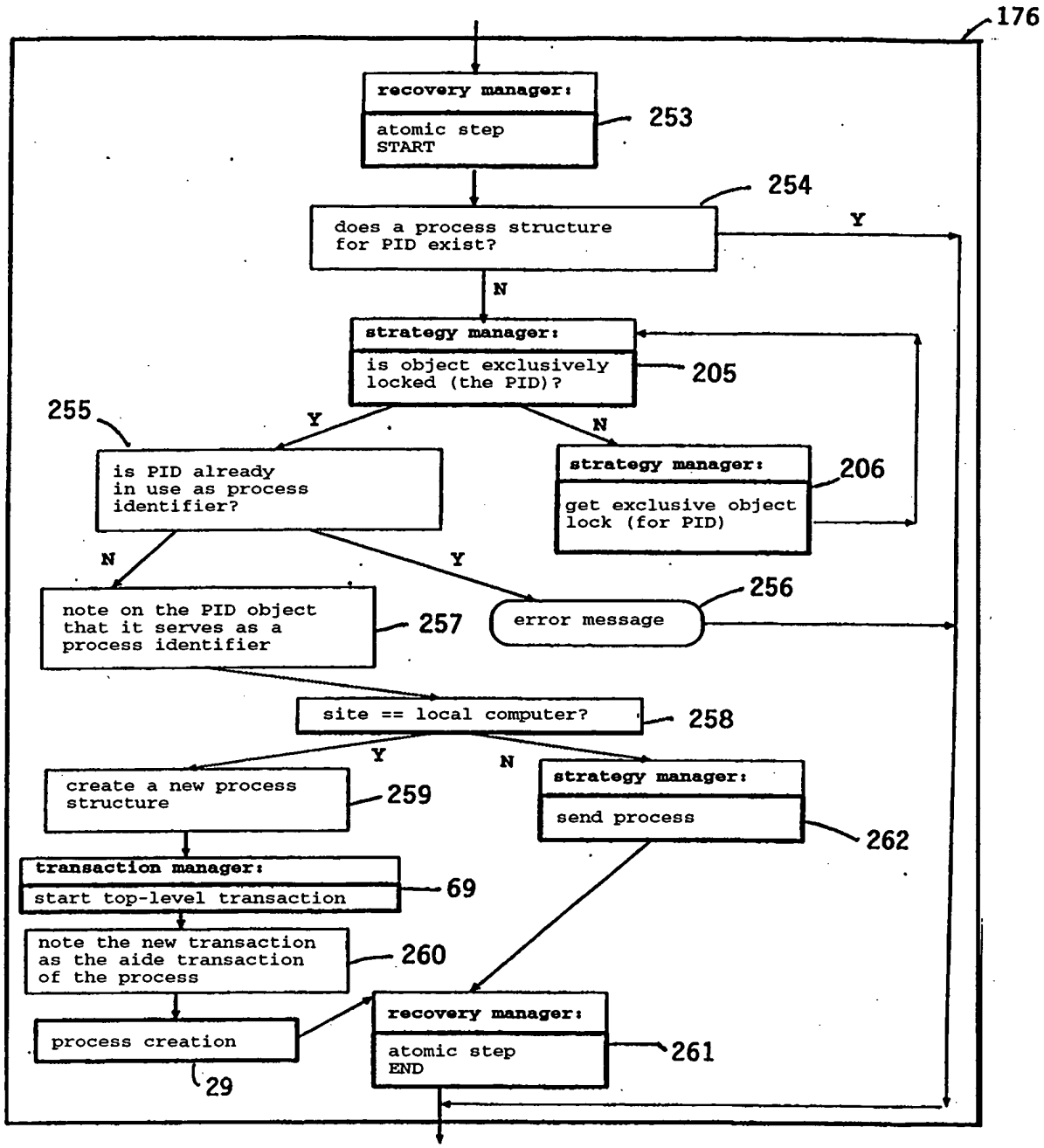


Fig.32

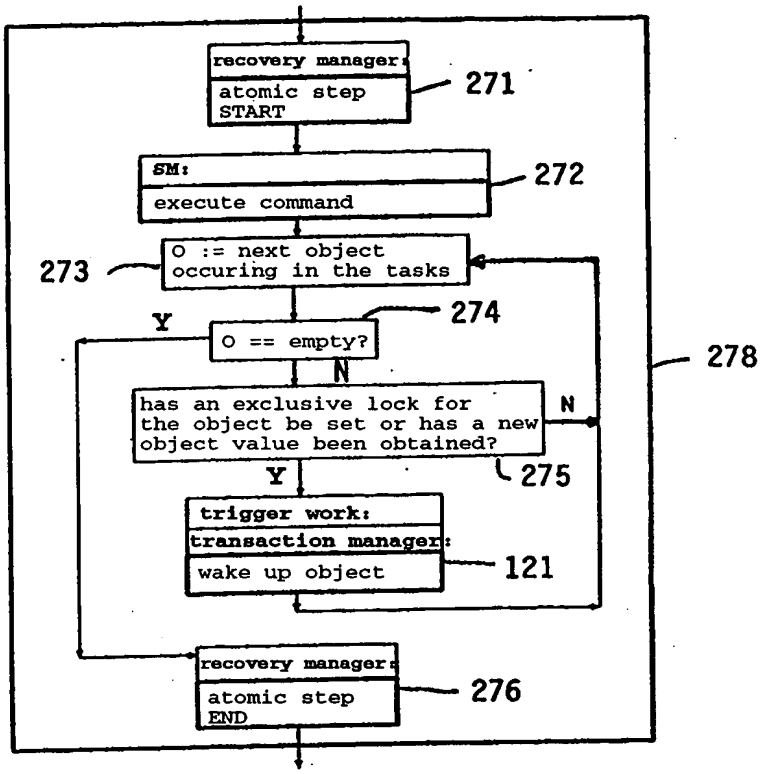

recovery manager:
atomic step
START
271
SM:
execute command
272
O := next object
occuring in the tasks
273
Y
O == empty?
274
N
has an exclusive lock for
the object be set or has a new
object value been obtained?
N
Y
275
trigger work:
transaction manager:
wake up object
121
recovery manager:
atomic step
END
276
278

Fig.33

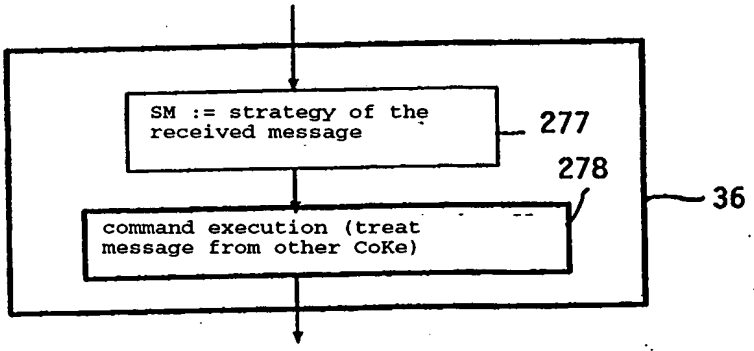

SM := strategy of the
received message
277
278
command execution (treat
message from other CoKe)
36

Fig.34

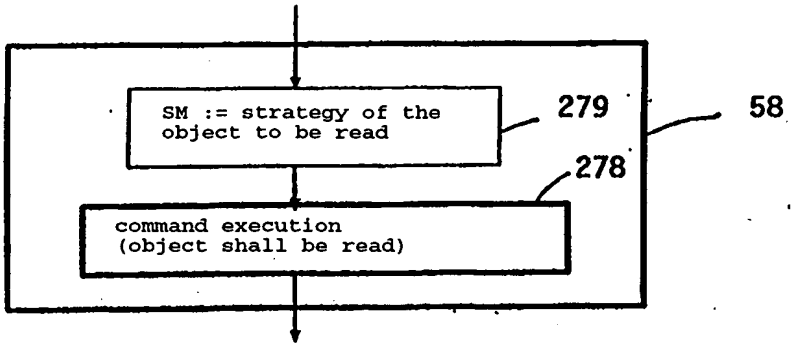


Fig.35

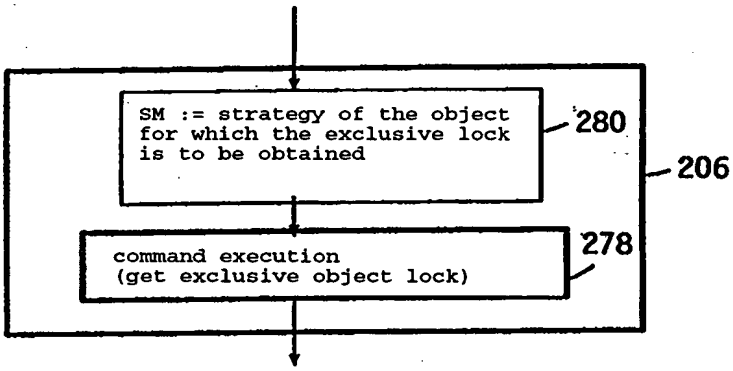
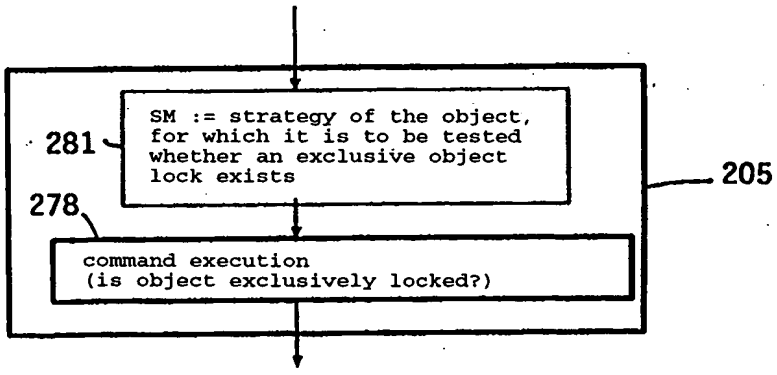


Fig.36



09/269485

Fig.37

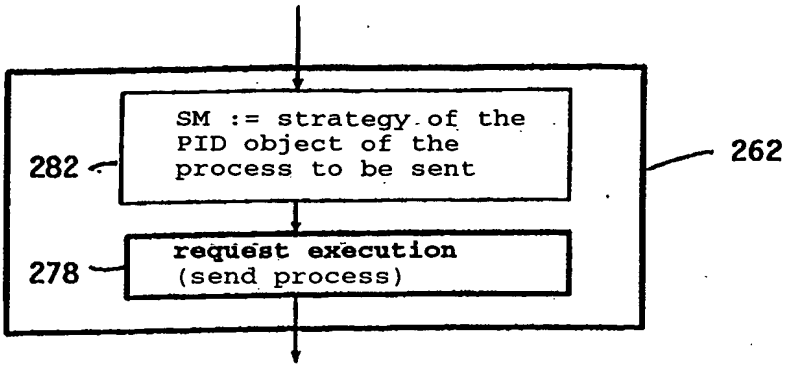


Fig.38

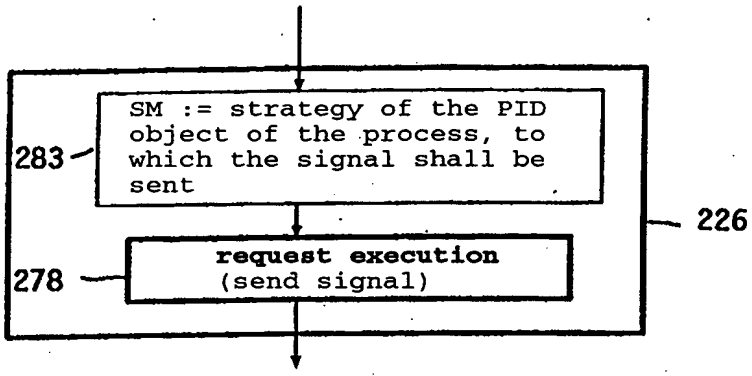


Fig.39

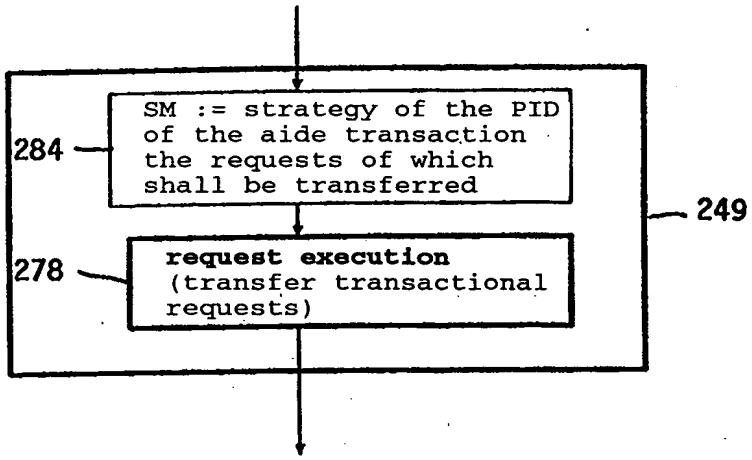
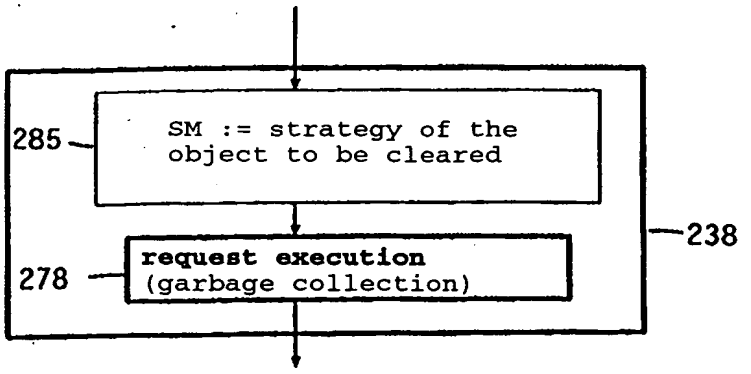



Fig.40